



WP

Lars Bækgaard

## Event-Based Conceptual Modeling

Informatics  
Research Group

# EVENT-BASED CONCEPTUAL MODELING

Lars Bækgaard<sup>1</sup>

## ABSTRACT

The paper demonstrates that a wide variety of event-based modeling approaches are based on special cases of the same general event concept, and that the general event concept can be used to unify the otherwise unrelated fields of information modeling and process modeling. A set of event-based modeling approaches are analyzed and the results are used to formulate a general event concept that can be used for unifying the seemingly unrelated event concepts. Events are characterized as short-duration processes that have participants, consequences, and properties, and that may be modeled in terms of information structures. The general event concept can be used to guide systems analysis and design and to improve modeling approaches.

## KEYWORDS

Business events, event-based modeling, process models, information models.

## 1. INTRODUCTION

The purpose of this paper is to present and discuss a conceptual model of business events as a foundation for event-based conceptual modeling. Many modeling approaches use events to support conceptual modeling of processes and/or information, but their underlying conceptualizations of events appear to be quite different and incompatible. There is no shared agreement about business events and their roles in information systems.

Business process modeling is a subject that has gained a lot of research attention recently (Raghu, Jayaraman et al. 2004; Danesh and Kock 2005; Sun, Zhao et al. 2006; Damij 2007; Frye and Gulledge 2007; Turetken and Schuff 2007; Wegmann, Lê et al. 2007). Many process modeling approaches are based on events. BPMN is a modeling language that views events as small processes, the occurrences of which are used to control process flows (White 2004). JSD (Jackson 1983) and OOAD (Mathiassen, Munk-Madsen et al. 2000) are methods that use events to represent shared actions. Event-activity diagrams use events to represent interaction between loosely coupled activities (Bækgaard 2004; Bækgaard 2007).

Many languages for information modeling do not support modeling of information about events directly. However, some information modeling approaches are based on events. In data warehouses, events can be used to represent and analyze information about business processes. The EVER model supports modeling of information about events in terms of networks of related events and entities (Bækgaard 1999). Some modeling approaches support integrated event-based modeling of processes and information. Action-oriented conceptual modeling uses actions (events) as the basis of activity modeling, and it uses entity-relationship diagrams to represent information about actions (Ågerfalk and Eriksson 2004). DEMO uses transactions (events) to represent business processes as chains of binary interactions and it uses facts to represent information about completed transactions (Dietz 2006). Action sequencing uses actions (events) as the basis of activity and information modeling

---

<sup>1</sup> Aarhus School of Business, University of Aarhus, Fuglesangs Alle 4, DK-8210 Aarhus V, Denmark, e-mail: lab@asb.dk.

(Andersen 2006). Event-based modeling uses events as the basis of both activity and information modeling (Bækgaard 1999; Bækgaard 2001).

The importance of events is immanent in the fact that events play important roles in many modeling approaches. Unfortunately, the approaches are based on seemingly different and incompatible event concepts. It is, however, possible to extract a conceptual event model from the approaches and to use the model to unify the event concepts of the approaches. The model characterizes events as short-duration processes which may be shared by a number of participants, which have consequences, which have properties and can be modeled in terms of information structures. The model characterizes an event as a short-durable process having significance for a business in the sense that it represents relevant information and/or affects the business processes. For example, when a business receives a customer order the business may collect relevant information about, say, the customer and the ordered items, and it may initiate activities like shipping and invoicing. Historical information about events can be used to build shared action memories that support actors participating in the activities.

The paper's major contributions are as follows. Based on an analysis of seemingly different event concepts, a conceptual event model is formulated. The analysis demonstrates that a broad range of event concepts can be viewed as special cases of the conceptual event model. The conceptual event model is used for demonstrating that events can integrate conceptual process and information modeling. This implies that events play central roles for the relations between processes and information. Events are parts of the activities in which actors use information about events. Events should be treated as first-class citizens that have process-like characteristics *and* information-like characteristics (Bækgaard 2001).

Section 2 describes the general notions of events and processes on which the paper is based. Section 3 analyzes a set of event-based modeling approaches, using an event concept that characterizes events in terms of participants, consequences, properties, and information. Section 4 discusses principles of event-based conceptual modeling based on the analysis in Section 3. Section 5 concludes the paper and suggests directions for future research.

## 2. EVENTS & PROCESSES

Processes are inherently related to change (Mathiassen 1984). The Latin word *processus* means movement. Change can be a movement in time and space, and it can be a change in the state of a phenomenon.

Some business processes are work activities in the sense that they are carried out by one or more actors performing actions (Checkland 1981). General business activities deal with movement, manipulation, and consumption of materials and information, and they deal with coordination in terms of requests for, agreements about, control of, and evaluation of work activities (Denning and Medina-Mora 1995). According to Davenport a business process is a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs (Davenport, 1993). According to Hammer and Champy a process is a collection of activities taking one or more kinds of input and creating an output that is of value to a customer (Hammer & Champy, 1993). A business process is a continuous series of organizational tasks, undertaken for the purpose of creating output (Ziemann, Matheis et al. 2007). Business activities have customers and they may cross organizational boundaries (Davenport and Short 1990; Ziemann, Matheis et al. 2007).

When a process is viewed as a set of work activities it is emphasized that one or more actors carry out the process. Sentences like “a borrower borrows a book at a library at a certain time” or “a customer orders an item from a store at a certain time” can be used to emphasize the roles of actors in activities. In the first sentence, the actor “borrower” carries out the activity “borrows”. In the second sentence the actor “customer” carries out the activity “orders”.

According to activity theory, human beings perform goal-directed actions directed towards objects (Leontiev 1978; Engeström 1991). The actions may be mediated by tools. An activity is a set of coherent actions that are directed towards a goal. The methods with which the action is accomplished are called operations. Operations are related to conditions. The ordering of a product via a website can be viewed as an activity directed towards the goal of possessing the product. The activity is comprised of actions like “select product” and “pay”. The actions are accomplished by operations that are related directly to the material characteristics of the website. For example, “select product” can be accomplished by mouse movements and keyboard typing.

The language-action perspective suggests that actions are more than material actions (Winograd and Flores 1986; Goldkuhl 2001). When human beings express themselves through language they are not merely communicating. Communicative actions may involve promises about future actions. When a business employee accepts a customer order, the acceptance is communicated to the customer. At the same time the acceptance involves a promise to ship the ordered products to the customer within some period of time.

Activity theory focuses mainly on material actions and the language-action perspective focuses mainly on communicative actions. Like other actions, communicative actions are accomplished by operations that are inherently material. When a library borrower uses a website to reserve a book, the action can be viewed as a communicative action that is accomplished via material operations related to the use of the website.

Some business processes are not activities performed by actors. They are not performed at all - they happen. Sentences like “blood flows”, “temperature rises” or “conveyor belt moves” emphasize that something is happening without the involvement of actors.

Some business processes are comprised of a combination of activities and non-activity processes. For example, actors may control (activity) chemical processes in a production plant, but the chemical processes (non-activity) happen. General business processes may be comprised of a combination of processes like mechanical processes, chemical processes, biological processes, and social processes.

A process can be represented by a description or a model in terms of, say data flow diagrams (De Marco 1978) or EPC diagrams (Keller and Teufel 1998; Dehnert 2002). Such diagrams represent processes understood as views on change. They do not represent the change, but they can be interpreted as views on change and as such be used to mediate conversations about change. It is never possible to describe all possible changes. Change is always perceived from a certain perspective. The term process denotes a view on a change rather than the change itself. Different processes represent different views on the changes of a phenomenon. Caverlee et al. view a process as a template that specifies dependencies between tasks. Such templates have instances following a pattern that is prescribed by the process and carried out by capable resources like human beings or software systems (Caverlee, Bae et al. 2007).

Events are short-duration processes representing moments in larger processes. Events can be viewed as significant occurrences with locations in time and space (Jackson 1983; Rumbaugh, Jacobson et al. 1999). When a phenomenon is viewed as an event it is assumed that a further subdivision of the phenomenon is unnecessary in a given context and that it makes sense to neglect its duration. Events occur within processes like chemical processes, biological processes, technical processes, or social processes. Phenomena like “customer files an order”, “borrower borrows a book”, “website user clicks”, “heart beats”, or “elevator starts moving” can be viewed as events. These phenomena have durations in time, but in certain contexts it makes sense to assume that they occur at points in time. An event is significant for a business if it makes a difference for the business, if the event occurs or not. An event is significant if its occurrence is worth remembering and/or if its occurrence affects other business processes. In most businesses the arrival of a customer order is treated as a significant event. When a customer order arrives, information about the customer and the products ordered may be

registered, and activities like packing, shipping, and invoicing may be initiated. An event is detectable if at least one actor is able to realize that the event has occurred. Events must be detectable either directly or indirectly via measuring apparatus. The notion of events being detectable and significant is similar to Bateson's notion of information as a difference that makes a difference (Bateson 1979). If an event is not detectable it does not make the first difference that brings the event to the attention of an actor. If an event is detectable, but not significant, it does not make the second difference that leads to action.

Like other processes, events do not exist per se. They are constructs that can be used to create orders in chaotic business processes. The ongoing negotiation about the business event space in a business is a construction of what makes sense and what is of significance in the business.

A process may express a continuous or discrete view on change. An event can be viewed as a discrete step in a process. As such it can be used to define the points in time at which a significant change occurs in the state of a system. A process represents a particular trajectory (or part thereof) in a system's state-transition space.

Events are double-sided phenomena. Events are processes *and* they are information. Events are parts of the processes in which actors use information about events. When events are viewed as information, attention is directed towards the activities in which actors participate and in which they may use action memories to improve the activities. This implies that events may play a central role for the relations between information and activity.

### 3. EVENT-BASED MODELING APPROACHES

This section describes a set of event-based modeling approaches. Some of the approaches are based on an explicitly stated event concept. Other approaches are based on concepts like actions and messages. These approaches will be interpreted to be event-based by interpreting actions and messages as events. All approaches will be interpreted in terms of the following aspects. Only the subset of these aspects that makes sense will be described for each approach.

*Participants.* Events have participants like human beings, technology and other artifacts (Bækgaard 2001; Poels, Maes et al. 2007; Alspaugh and Antòn 2008). Does a particular modeling approach support modeling of participants? How does it support this modeling?

*Consequences.* Events may have consequences (Dehnert 2002; White 2004; Lübke, Lücke et al. 2006). Does a particular modeling approach support modeling of consequences? How does it support this modeling?

*Properties.* Events may have descriptive properties (Bækgaard 2001; Bækgaard 2004). Does a particular modeling approach support modeling of properties? How does it support this modeling?

*Information.* Events can be modeled in terms of information structures (Bækgaard 1999; Bækgaard 2001; Ågerfalk and Eriksson 2004; Andersen 2006). Does a particular modeling approach support modeling of information about events? How does it support this modeling?

Siau and Rossi have identified three ways of evaluating modeling methods. Feature comparison is based on a checklist of features against which a set of modeling methods are characterized. Theoretical/conceptual investigation is based on concepts or metrics that are used to interpret a set of modeling methods. Empirical evaluation is based on investigation methods like surveys, experiments, case studies, action research, and verbal protocol (Siau and Rossi 2007).

The method that is used to analyze modeling approaches in this section can be characterized as a combination of a feature list approach and a theoretical/conceptual approach. The four aspects (participants, consequences, properties, information) can be viewed as an ontology in terms of which the approaches are interpreted. At the same time they can be viewed as a feature list that is used to characterize the modeling approaches.

### 3.1. Event-less approaches

Many process modeling approaches are event-less in the sense that they are not based on explicit notions of events. This does not imply that these languages cannot be used to support event modeling. However, it implies that the modeler must use and interpret non-event concepts as if they were event concepts. Examples are conceptual models (Checkland 1981), data flow diagrams (De Marco 1978), ISAC flow graphs (Lundeberg, Goldkuhl et al. 1978), and activity diagrams (Rumbaugh, Jacobson et al. 1999), and sequence diagrams (Rumbaugh, Jacobson et al. 1999).

Many information modeling approaches do not support event-modeling directly. They are based on concepts like records, objects, or surrogates. Record-based modeling languages use records - sequences of values - as basic modeling constructs. Examples are the relational model (Codd 1970) and nested relational models (Thomas and Fischer 1986; Abiteboul, Fischer et al. 1989). Object-based modeling languages use objects as their basic modeling constructs. An object is composed of an identifier, a set of values, and a set of methods. The values represent the object's state. The identifier identifies the containing object uniquely within a set of objects. The methods define actions that the object can perform. Object-based structures can be modeled by means of class diagrams (Rumbaugh, Jacobson et al. 1999). Surrogate-based modeling languages use surrogates as their basic modeling constructs. A surrogate is a modeling element which itself has no visible properties apart from its mere existence. The basic idea is that every entity of the outside world is represented by a surrogate in a model (Hall, Owlett et al. 1976). Examples are the functional model (Shipman 1981), the extended relational model (Codd 1979), Telos (Mylopoulos, Borgida et al. 1990; Mylopoulos 1992), the semantic data model (Hammer and McLeod 1981), and the entity-relationship model (Chen 1976). The object-role model (ORM) is a language that supports modeling of information in terms of objects and roles played by objects (Halpin 2001). The roles are intended to reflect the structure of natural language sentences like "borrowers participate in borrow events".

### 3.2. EVER diagrams

EVER (event-entity-relationship) diagrams can be used to represent information in terms of related events and entities (Bækgaard 1999; Bækgaard 2001).

*Interpretation.* An EVER event is an occurrence with a negligible duration.

*Information.* EVER diagrams can be used to represent information about events in terms of event histories. Each event can be related to entities that represent participants and to values that represent properties.

*Related approaches.* In data warehouses events can be used to represent information about business activities that can be stored and analyzed. For example, a data warehouse may store information about customer behavior in terms of information about selected events involving customers. Among such events may be order, pay, and deliver. The functional model is an information modeling language in which information is represented by entities and functions that map entities to entities or values (Shipman 1981). Orman has demonstrated how the functional model can be used to represent information about events in terms of event histories (Orman 1988).

### 3.3. Atomic database transactions

An atomic database transaction is an all-or-nothing procedure that modifies a database (Gray and Reuter 1993). Database transactions correspond to events like ordering of products or services and they are used to update databases when such events occur. For example, an order transaction can be used to update a database to reflect the fact that a particular customer has ordered a particular product.

*Interpretation.* An atomic transaction is a short-duration process that can be interpreted as an event.

*Consequences - state.* A database is updated as the consequence of an event (transaction).

*Properties.* An event has properties if the corresponding transaction has parameters.

### 3.4. State charts

A state chart can be used to model a state dependent phenomenon in terms of rules for the sequencing of messages sent to and from a state dependent object (Rumbaugh, Jacobson et al. 1999). A state chart represents such an object.

*Interpretation.* State charts are based on named messages with parameters. Such a message can be interpreted as an event with the same name as the message and with a set of properties that correspond to the parameters of the message.

*Consequences - state.* An event (message) causes a state-transition. For example, a state chart that represents a library book changes state from “ready” to “borrowed” when a borrow event (message) occurs.

*Consequences - process.* An event (message) triggers a set of actions. For example, a monitoring procedure can be triggered when a library book is borrowed. The procedure monitors that the book is returned in a timely manner according to the policies of the library.

*Properties.* An event have properties that are embedded in the parameters of the corresponding message.

*Related approaches.* Active databases are based on ECA (Event-Condition-Action) rules (Widom and Ceri 1996). When a certain event occurs and if a specified condition is true, then a specified set of actions are executed. For example, this can be used to ensure that proper database transactions are executed when certain events occur.

### 3.5. JSD

The development method JSD (Jackson System Development) uses structure diagrams to model entity life cycles (Jackson 1983). A structure diagram corresponds to an entity type. It defines a set of legal sequences of actions that can be performed or that influence entities of the particular type.

*Interpretation.* JSD is based on actions. A set of actions can be interpreted as a composite action that can be interpreted as an event in which a set of entities participate. For example, a borrower and a book play roles when a book is borrowed in a library. The borrower performs a borrow action and the book is influenced by a borrow action. These two roles can be interpreted as a shared event representing both the borrower’s and the book’s roles.

*Participants.* Events have multiple interacting participants. An event is initiated by an actor. This implies that entity life cycles are based on an event perspective utilizing the fact that events can have multiple interacting participants.

*Consequence - process.* An entity life cycle represents the legal evolution of a certain entity type in terms of rules that constrain the events in which the entities participate. The occurrence of certain events influences the admissibility of future events. When a borrow event occurs, other borrow events with the same book as participant are disabled and one return event with the same book as participant is enabled. All entities that participate in an event must be ready to participate. The life cycles of the participants must interact. Events are not related to larger business activities. Consequently, events cannot be used to trigger or terminate general business activities. Likewise, events cannot trigger events.

*Properties.* An event has a set of descriptive properties like quantities and dates.

*Related approaches.* A number of life cycle modeling approaches that resemble JSD have been proposed (Rosenquist 1982; Kappel and Schrefl 1991; Avison and Fitzgerald 2006). OOAD is a

development method that uses state charts to define object life cycles in the same manner as entity life cycles are defined in JSD (Mathiassen, Munk-Madsen et al. 2000).

### 3.6. EPC diagrams

EPC (Event-driven Process Chains) diagrams can be used to represent business processes in terms of events, functions, and logical operators (Keller and Teufel 1998; Dehnert 2002). An event is a named state change that is used to control the flow of activity. A logical operator is used to branch or merge the flow of activity. The selection of a the particular sequence depends on the state changes (represented by events) that occur.

*Interpretation.* An EPC event represents a named state change. More precisely, it represents the entering of a state.

*Consequences - process.* Events trigger business activities (functions) or they are the result of the finishing of an activity. Events can be used to control the flow of sub-activities.

### 3.7. BPMN diagrams

BPMN (Business Process Modeling Notation) diagrams support business activity modeling in terms of events and activities (White 2004). BPMN supports process modeling in terms of sequences, alternations, and iterations of activities. A modeled process can be organized in terms of swim lanes and pools. Swim lanes can be used to represent strongly coupled actors that can influence each other's process flows. Pools can be used to represent loosely coupled actors that interact in terms of exchange of information.

*Interpretation.* BPMN events are non-durable processes that affect the flow of a process.

*Consequence - process.* When an event occurs the process flow is changed. Events can be used to control the flow of sub-activities. Events can be used to trigger and interrupt activities. BPMN events cannot be used to model interaction between multiple participants.

*Properties.* An event may have an associated occurrence time.

### 3.8. Event-activity diagrams

An event-activity diagram defines a type of activity and it is composed of triggering events, interrupting events, and an activity (Bækgaard 2004).

*Interpretation.* Event-activity diagrams are based on events that are assumed to have negligible durations. An event can represent shared actions.

*Participants.* Event-activity diagrams use events with participants to model interaction between two or more concurrent activities. Events with multiple participants represent time points where multiple activities must interact. The event-activity diagrams in Figure 2 interact via the shared events Borrow and Return. For example, when a Borrower instance requests a Borrow event the event is executed if and only if the corresponding BookCopy instance is ready to participate. An event can have any number of participants.

*Consequences - process.* An activity is initiated when a triggering event occurs. The activity runs until it reaches an exit point or until an interrupting event occurs. One or more actors carry out the activity. An event changes the flow of business activities. An event-activity diagram defines an activity type that can have multiple, concurrent instances. An instance of an activity type is generated by the occurrence of an event that conforms to a triggering event type. The subdivision of activities into concurrent, interacting activities facilitated by this mechanism makes it possible to define activities in a distributed manner resembling the distribution of activity among actors in a business.

### 3.9. Resources-events-agents diagrams

Resources-events-agents diagrams represent events as short-duration processes in which agents participate. Resources flow to and from events (Poels, Maes et al. 2007).

*Interpretation.* Events are short-durable processes.

*Participants.* Events have a number of participants. Participants can be human beings, agents and material resources.

### 3.10. Essential events

The development method Modern Structured Analysis uses essential events to trigger essential activities (Yourdon 1989; Yourdon 2003). An essential activity is a coherent business activity that is modeled by a subset of a data flow diagram. The activity is triggered by the occurrence of a corresponding essential event. For example, an essential activity may represent the activity that is triggered when an order is received.

*Interpretation.* Events are non-durable processes.

*Consequences.* When an essential event occurs, the corresponding essential activity is activated.

*Related approaches.* Use cases are similar to essential activities in the sense that an activity is triggered by the occurrence of an event. A use case represents a coherent unit of functional behavior that is offered by a system to one or more actors (Jacobson, Booch et al. 1999). A use case defines the behavior of a subsystem and it defines the subsystem's interaction with one or more actors. The behavior is initiated by an event (action) performed by a primary actor.

### 3.11. Action-oriented conceptual modeling

Action-oriented conceptual modeling is a modeling approach that is based on the language-action perspective (Ågerfalk and Eriksson 2004). Fundamentally, the idea is to use actions as the basic units of process and information modeling.

*Interpretation.* Action-oriented conceptual modeling is based on actions that can be interpreted as events.

*Participants.* Events (actions) have a number of participants. For example, an event can be described as "employee sends invoice via system to customer". Such an event has four participants (employee, invoice, system, customer).

*Consequences – process.* Action diagrams (Goldkuhl 1996) are used to represent a set of actions and their inputs, outputs, and temporal dependencies. Such diagrams define rules for the flow of communicative and/or material actions in terms of conditions for the execution of certain actions.

*Consequences – states.* Action diagrams can be annotated with symbols for named states that are the consequences of events (actions). For example, the state after an order is received may be called "order received".

*Information.* Entity-relationship diagrams (Chen 1976) are used to represent information about events and their related entities.

### 3.12. DEMO

DEMO (Dynamic Essential MOdeling) is a framework that supports modeling of interaction between independent actors (Dietz 2006). Like action-oriented conceptual modeling, DEMO is based on the language-action perspective. DEMO uses business transactions to represent business processes and it uses facts to represent successful transactions. A business transaction is an activity with three phases. In the order phase an initiator and an executor agree on the delivery of a product or a service. In the

execution phase an executor creates the product needed and/or performs the service needed. In the result phase the initiator and the executor negotiate about the success of the outcome. For example, a customer (initiator) may request a certain product (order phase), an employee (executor) finds the product for the customer (execution phase), and the employee acknowledges that the product is as desired (result phase).

*Interpretation.* DEMO is based on transactions. A transaction can be interpreted as an event that represents a set of interactions between the initiator and the executor.

*Participants.* An event (transaction) has two actors (the initiator and the executor) as participants.

*Consequences - process.* A successful event (transaction) has consequences in the sense that it changes the set of potential future events. And it has consequences in the sense that the initiator gets the requested product or service.

*Properties.* An event (transaction) may have any number of properties.

*Information.* When an event has been completed successfully, the state of the corresponding business has changed. This change is represented by a fact, the structure of which corresponds to the completed event. The set of potential facts, corresponding to an event type, is characterized by a fact type.

### 3.13. Action sequencing

Action sequencing is a modeling approach that supports definition of rules for sequencing of actions (Andersen 2006). It is a linguistic approach that is based on an analysis of structures of linguistic expressions that refer to human activity (Fillmore 1968; Fillmore 1977; Nurcan, Etien et al. 2005). Action sequencing highlights the relations between information modeling and process modeling even though its primary purpose is process modeling. The basic idea is that an action type is presented by a linguistic sentence structure with unfilled roles that can be played by, say human beings, things, and representations.

*Interpretation.* Action sequencing is based on actions that can be interpreted as events.

*Participants.* An event (action) type has a set of roles that can be filled by participating human beings, IT systems, things, and information.

*Consequences - process.* Types of actions are defined by means of sentence structures, and dynamic relations between sentence structures are used for defining a rule space within which events (actions) occur. Dependencies between sentence structures can be defined in order to express constraints on the potential participants that are allowed to play roles in specific situations. For example, a sentence expressing that John borrows a specific book at a library must at some point in time be followed by a sentence expressing that John returns the book.

*Properties.* An event (action) may have any number of properties.

*Information.* A sentence can be viewed as a fact that conforms to a sentence structure. A sentence structure can be viewed as a schema and a sentence can be viewed as an instance of such a schema. Consequently, sentence structures can be viewed as basic units of both activity models and information models. Sentences and DEMO facts are closely related.

### 3.14. Event-based modeling

Event-based modeling is a modeling approach that views events as double-sided occurrences (Bækgaard 2001). Besides, events are change agents, i.e. activities with multiple participants. Events can be modeled as activities with multiple participants in terms of event-activity diagrams in which events can be used to synchronize the activities of their participants.

*Interpretation.* Event-based modeling is based on non-durable events, the types of which are expressed as signatures on the form Name [Participants] (Properties).

*Participants.* An event may have any number of participants.

*Consequences - process.* Event-activity diagrams are used to represent activities.

*Properties.* An event may have any number of properties.

*Information.* EVER diagrams are used to represent information about events.

### 3.15. Discussion

The analyzed modeling approaches can be grouped into three types of approaches. Information modeling approaches focuses solely on information modeling (Orman 1988; Bækgaard 1999; Bækgaard 2001). Process modeling approaches focus solely on process modeling (Gray and Reuter 1993; Rumbaugh, Jacobson et al. 1999; Dehnert 2002; Bækgaard 2004; White 2004). Integrated approaches focus on integrated process and information modeling (Bækgaard 2001; Ågerfalk and Eriksson 2004; Andersen 2006; Dietz 2006). The approaches are based on different aspects of events like, say, a statement of the fact that the event has occurred, a short-duration process occurrence, or a named state change. In the following section these views will be integrated in terms of an event ontology.

## 4. PRINCIPLES OF EVENT-BASED CONCEPTUAL MODELING

The preceding section summarizes an analysis of a set of event-based approaches using events in different ways to control processes and as sources of information. The analysis indicates that the approaches are based on variants of a more general event concept. Even though they are based on seemingly different event concepts it is possible to outline an event ontology that covers all the event concepts in terms of the following four aspects: participants, consequences, properties, and information.

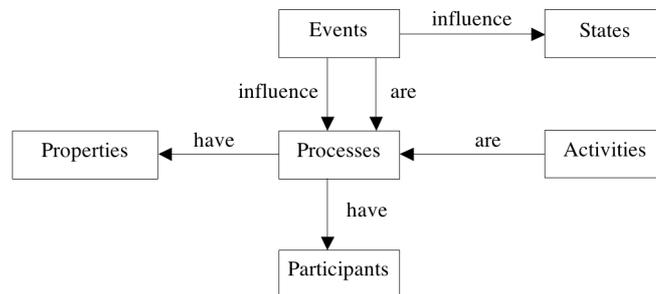


Figure 1 - Event-based ontology

Figure 1 outlines an event ontology that is based on the analysis of modeling approaches in the preceding section. Events are processes. Some events are activities. Events have properties like dates, times, and quantities. Events have participants like human beings, things, and representations. Events influence states like the states of businesses or the states of databases. Events influence processes in the sense that they, say, trigger or interrupt activities.

Events are short-duration processes that play important roles in business processes in which actors act and in which they may use action memories to improve the actions. Events affect business processes and they represent information that informs actors. This implies that events may play central roles for the relations between information and activity. Events cause state changes, they control processes, and they represent interaction between two or more loosely coupled processes.

The following subsections characterize the following four aspects of events. They may be shared by a number of participants, they have consequences, they may have properties, and they can be modeled in terms of information structures. The four aspects have been identified by means of the analysis, the results of which are described in Section 3.

#### 4.1. Participants

Real-world activities are composed of loosely-coupled, interacting, concurrent activities that interact from time to time. For example, the activities of customers and sales persons must interact when a customer buys an item. At other times the activities of customers and sales persons are mutually independent. Global, loosely coupled processes like cross-organizational business processes make it necessary to focus on process interaction (Ziemann, Matheis et al. 2007). For example, the supply-chain of a global business may be comprised of many scattered and autonomous processes that must interact.

All events are processes. Some events are activities and as such they can be viewed as shared actions performed by actors. Some events are not activities. All events have participants like human beings, IT systems, machines, things like produced items, or information like sales reports. At least one actor participates if an event is an activity. Borrow events in a library may have borrowers and books as participants.

An event is shared if it has two or more participants. A shared event can be used to represent a named time point in which actors, things, and information meet and interact. For example, an ordering event can be viewed as a time point where a customer and a product meet. Ordering events may have participants like customers and products. Events can be used to represent interaction between processes that are carried out by multiple, relatively independent participants (actors and objects.) A subdivision of processes into concurrent, interacting processes makes it possible to define processes in a distributed manner that resembles the distribution of processes among actors in a business.

Event-activity diagrams (Bækgaard 2004) use shared events to represent interaction between loosely coupled processes as illustrated in Figure 2. Structure diagrams (JSD) can be used in a similar but more restricted manner to represent interaction between loosely coupled processes (Jackson 1983). Action-sequencing (Andersen 2006), DEMO (Dietz 2006), and event-based modeling (Bækgaard 2001) use sentence-like expressions to represent events with interacting participants.

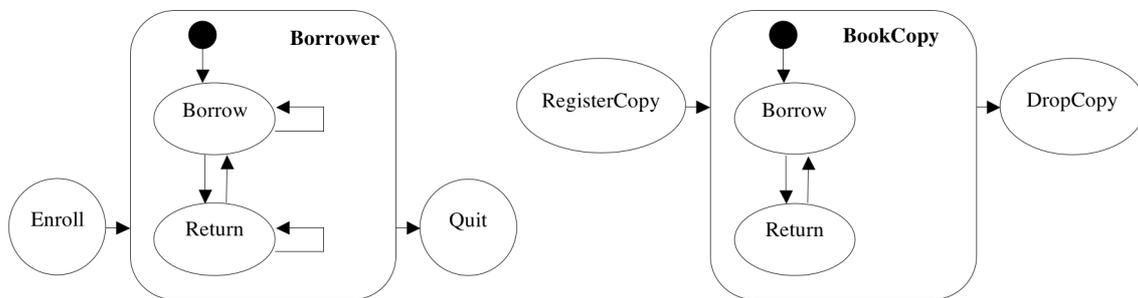


Figure 2 - Event-activity diagrams

The event-activity diagram labeled Borrower in Figure 2 models activity related to borrowers in a library. A Borrower activity is triggered by an Enroll event and it is interrupted by a Quit event. A Borrower activity is a sequence of Borrow and Return events. The diagram labeled BookCopy models activity related to book copies in a library. A book copy is a physical copy of a certain book title. A BookCopy activity is triggered by a RegisterCopy event and it is interrupted by a DropCopy event. A BookCopy activity is composed of an alternating sequence of Borrow and Return events.

#### 4.2. Consequences

Events have consequences. When they occur, something is changed. The changes may take different forms. Some changes are related to changes in the state of systems. Other changes are related to changes in business processes.

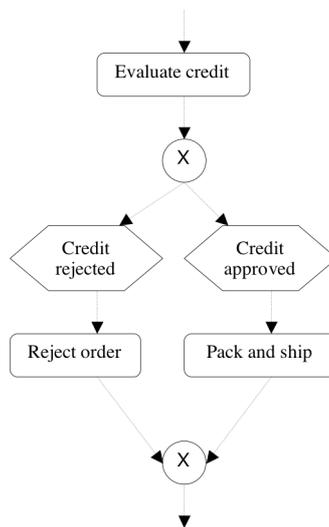
*States*

An event can change the state of the surrounding business. In a library the result of a borrow event may be that the state of the borrowed book changes from “available” to “borrowed”. Likewise, the result of a return event may be that the returned book changes from “borrowed” to “available”. When a physical product is sold and shipped the current stock is reduced. A “deliver ordered product” event influences the state of a business by reducing the number of available products and by changing a product's database in order to record the reduced number of available products.

State charts can be used to model state changes that are caused by events (Rumbaugh, Jacobson et al. 1999). For example, a library book may change state from “available” to “borrowed” as a consequence of a “return book” event. EPC diagrams represent events by named state changes (Keller and Teufel 1998; Dehnert 2002). For example, the fact that an application changes state to “approved” may be represented by an EPC event. Action-oriented conceptual modeling uses action diagrams to represent business actions (Ågerfalk and Eriksson 2004). Such diagrams can be annotated with named states that are the results of business actions.

*Processes*

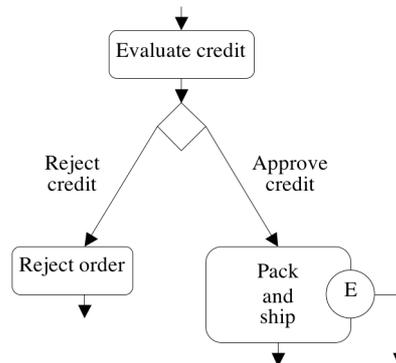
An event influences business processes (Caverlee, Bae et al. 2007; McGinnis 2007). An event may enable future processes. When a borrowed book is returned to a library, it is ready to be borrowed by other borrowers. An event may represent a commitment for future processes in terms of necessary future actions. When a customer orders an item in a business the business commits to deliver the item within a specified future. Events trigger processes. When a book is borrowed in a library, a process monitoring timely returning of the book may be initiated. When a business receives an order one or more employees may carry out packing and shipping activities, and an ERP system may carry out invoicing activities. A “deliver ordered product” event influences various business activities. For example, an order may be sent to a supplier if the stock is below some threshold, and an invoicing process may be triggered. Events may be constrained in the sense that participants may not be ready to participate in requested events at certain points in time. A book that is currently borrowed cannot participate in a borrowing event. Events terminate activities. A search activity may be terminated if a time limit is reached.



**Figure 3 - EPC diagram**

The EPC diagram in Figure 3 represents parts of an order processing and it is interpreted as follows: The activity is initiated by the function “Evaluate credit” that is followed by an exclusive split. Either

credit is requested or payment is rejected (but not both). Either “Reject order” or “Pack and ship” has been executed (but not both).



**Figure 4** - Partial BPMN diagram

The partial BPMN diagram in Figure 4 is based on the same example as the EPC diagram in Figure 3. The diagram illustrates that events (circles) can be used to control the flow of a process in a manner that is similar to EPC events. In particular, the event E defines a termination criterion for the activity “Pack and ship”.

Essential events trigger processes (Yourdon 1989; Yourdon 2003). Event-activity diagrams use events to trigger and interrupt processes (Bækgaard 2004). BPMN diagrams use events to trigger and interrupt processes (White 2004). EPC diagrams use events to trigger processes (Keller and Teufel 1998; Dehnert 2002). State charts use events to trigger actions (Rumbaugh, Jacobson et al. 1999).

### 4.3. Properties

Events have descriptive properties like time and quantities. A property can be a value like the text string “Joe” or the number 7124. For example, ordering events may have properties like quantities, dates and times. All events have occurrence times. JSD (Jackson 1983), state charts (Rumbaugh, Jacobson et al. 1999), DEMO (Dietz 2006), action sequencing (Andersen 2006), and event-based modeling (Bækgaard 2001) are based on events having any number of descriptive properties.

### 4.4. Information

Information about events and their consequences must be recorded in order to support business actors' decisions and activities. In general, information about the phenomena in Figure 1 and their mutual relations can be recorded in terms of a combination of information about states and information about event histories.

Actors produce and use information when they participate in business processes (Checkland and Holwell 1998). This is one of the reasons why most IT based information systems contain one or more databases. And it is one of the reasons why information about events plays a central role in enterprise information systems. Many such systems are based on databases in which information is stored and from which information is retrieved.

Information about an event may be represented directly in terms of the event's properties and participants, and it may be represented indirectly in terms of the effect of the event. Information models can be expressed in terms of, say, entity-relationship diagrams (Batini, Ceri et al. 1992) or class diagrams (Booch, Rumbaugh et al. 1999). Such diagrams can be interpreted as definitions of general properties of the modeled information *and* as conceptual definitions of the modeled phenomena.

Information about events can be stored in terms of event histories. The basic idea is to store information about all historical events and their participants and properties. Such information can be used to represent the process memory of a business. The purpose is to create a shared pool of information to be used by actors to improve the processes in which they participate. For example, a data warehouse with multi-dimensional information about customer behavior may be based on information about events like ordering and payment (Kimball 1996; Bækgaard 1999).

EVER diagrams can be used to represent an organization's process memory and other relevant information (Bækgaard 1999). Goldkuhl & Ågerfalk use the term action memory in order to emphasize that information about historical actions may be prerequisites for new actions (Goldkuhl and Ågerfalk 1998). An EVER diagram represents information about historical, current or future states and events. Such a diagram supports full temporal modeling in terms of existence intervals for entities and time point values for events. Event-based information structures combine the characteristics of multi-dimensional modeling of historical information in data warehousing environments with conventional entity-relationship modeling.

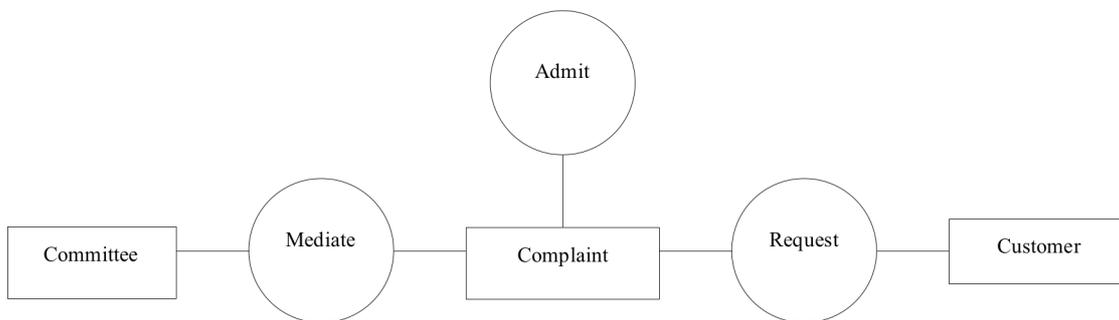


Figure 5 - EVER diagram

The diagram in Figure 5 contains three entity sets (Committee, Complaint, Customer) and three event sets (Mediate, Admit, Request). A new event is added to Request and a new entity is added to Complaint each time a customer requests a complaint. Entities and events can be related to values that are interpreted as attributes. Temporal attributes are unnecessary because the EVER model has built-in temporal support.

Information about the effect of events can be modeled by means of, say, entity-relationship diagrams (Batini, Ceri et al. 1992) or class diagrams (Booch, Rumbaugh et al. 1999). For example, the status of orders may be changed as the results of certain events. Information about the changed status may be stored directly as attributes of entities that represent orders. Entity-relationship diagrams represent information about historical, current, or future states.

DEMO (Dietz 2006), Event-based modeling (Bækgaard 2001), and Action sequencing (Andersen 2006) can be used to model information about events in ways that are based on the structure of event occurrences. DEMO uses fact types to express event (transaction) types. Event-based modeling uses event signatures to define event types. Action sequencing uses sentence structures to define event types. The EVER diagram in Figure 5 can be interpreted as a representation of the following event signatures that use the notation of event-based modeling.

- Admit [Complaint]
- Mediate [Committee, Complaint]
- Request [Customer, Complaint]

These event signatures can easily be formulated in the notations of DEMO (fact types) and action sequencing (sentence structures). Event signatures and sentence structures express events with an arbitrary number of participants. DEMO's fact types are, however, limited to events that have two actors as participants at most.

#### 4.5. Event types

Events are short-duration processes with participants, consequences, and properties. As argued in Section 2 processes (and hence activities) do not exist in any absolute sense. A process represents a view on change. Events can be used to represent change in a system with a state-transition space. Events play central roles in such systems because they are closely related to the systems' state-transition spaces. One or more events must occur in order for a system to change state. Therefore, identification of relevant events is crucial for a system's ability to have a rich state-transition space.

Events can be used to analyze and design business processes. A business system can be characterized by the space of events that may occur and by the events that actually do occur as time passes. Events are parts of the processes in which actors use information about events. Events can be used to organize business processes. First, they represent points in time in which the future direction of a business process is determined. Second, they represent points in time in which relevant information can be observed and recorded.

A candidate event type with names, properties, and participants must be characterized in terms of its effects on states and processes. Do occurrences of events of the given type change the state of a business and/or its environment in a significant manner? Does it influence other processes in a significant manner?

A candidate event type should be denoted by a name that describes the intention of occurrences of events of the particular type. Each candidate event type should have a set of properties like date, times, quantities, and other forms of descriptive properties that can be used to characterize occurrences of events of the particular type.

A candidate event type should have an associated set of participant roles that defines who and/or what constitutes legal and relevant participants. For example, an event type named "customer orders product" may have properties like date, quantity of the ordered product, payment method, and shipping address. The event type may have associated participant roles like customer and product. An occurrence of the event type should have specific content in date, quantity, payment method, and address. And it should be associated with a reference to a specific customer and to a specific product.

Event types can be expressed as event signatures, sentence structures, or fact types as illustrated in the example in the preceding subsection. And they can be represented graphically as event-based information structures as illustrated in Figure 5.

### 5. CONCLUSION

Events are short-duration, double-sided phenomena that can be viewed as both process and information. Events have process-like characteristics in the sense that something is changed when an event occurs. Events can be used to control the flow of processes. Events can be used to trigger and terminate processes and they can be used to enable and disable processes. Events have information-like characteristics in the sense that they can be observed and described. Information about events and their consequences can be registered, stored, manipulated, and presented for actors.

Many approaches support event-based modeling of processes and information. Even though they are based on seemingly different event concepts it is possible to outline an event ontology that covers all the approaches' event concepts in terms of the following four aspects: Participants, consequences, properties, and information. The event ontology in Figure 1 is based on these aspects.

Future work includes experiments with event-based modeling in support of business service modeling. Business services are characterized by many types of interaction between actors and various types of resources, and it is likely that events can be used to conceptually represent such interaction.

## 6. REFERENCES

- Abiteboul, S., P. C. Fischer, et al., Eds. (1989). *Nested Relations and Complex Objects in Databases*. Lecture Notes in Computer Science, Springer-Verlag.
- Ågerfalk, P. J. and O. Eriksson (2004). "Action-Oriented Conceptual Modelling." *European Journal of Information Systems* **13**(1): 80-92.
- Alspaugh, T. A. and A. I. Antòn (2008). "Scenario Support for Effective Requirements." *Information and Software Technology*(50): 198-220.
- Andersen, P. B. (2006). "Activity-Based Design." *European Journal of Information Systems* **15**: 9-25.
- Avison, D. E. and G. Fitzgerald (2006). *Information Systems Development. Methodologies, Techniques & Tools*, Blackwell Scientific Publications.
- Bateson, G. (1979). *Mind and Nature: A Necessary Unity*, Bantam Books.
- Batini, C., S. Ceri, et al. (1992). *Conceptual Database design. An Entity-Relationship Approach*, Benjamin/Cummings.
- Bækgaard, L. (1999). *Event-Entity-Relationship Modeling in Data Warehouse Environments*. International Workshop on Data Warehousing and OLAP (DOLAP'99). Kansas City, USA.
- Bækgaard, L. (2001). *Event Modeling*. Information Modeling in the New Millennium. M. Rossi, Siau K., Idea Group Publishing.
- Bækgaard, L. (2004). *Event-Based Activity Modeling*. Conference on Action in Language, Action in Language, Organisations and Information Systems (ALOIS'04). Linköping, Sweden.
- Bækgaard, L. (2007). *Event-Based Information System Models*. 9th International Conference on Enterprise Information Systems (ICEIS'07). Funchal, Madeira, Portugal.
- Booch, G., J. Rumbaugh, et al. (1999). *The Unified Modeling Language User Guide*, Addison-Wesley.
- Caverlee, J., J. Bae, et al. (2007). "Workflow Management for Enterprise Transformation." *Information Knowledge Systems Management* **6**: 61-80.
- Checkland, P. (1981). *Systems Thinking, Systems Practice*, Wiley.
- Checkland, P. and S. Holwell (1998). *Information, Systems, and Information Systems*, Wiley.
- Chen, P. P. (1976). "The Entity-Relationship Model - Towards a Unified View of Data." *ACM Transactions on Database Systems*.
- Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks." *Communications of the ACM*.
- Codd, E. F. (1979). "Extending the Database Relational Model of Data to Capture More Meaning." *ACM Transactions of Database Systems* **4**(4): 397-434.
- Damij, N. (2007). "Business Process Modelling Using Diagrammatic and Tabular Techniques." *Business Process Management Journal* **13**(1): 70-90.
- Danesh, A. and N. Kock (2005). "An Experimental Study of Process Representation Approaches and their Impact on Perceived Modeling Quality and Redesign Success." *Business Process Management Journal* **11**(6): 724-735.
- Davenport, T. H. and J. E. Short (1990). "The New Industrial Engineering: Information Technology and Business Process Redesign." *Sloan Management Review*: 11-27.
- De Marco, T. (1978). *Structured Analysis and System Specification*, Prentice-Hall.
- Dehnert, J. (2002). *Making EPCs Fit for Workflow Management*. Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK'02). Trier, Germany.
- Denning, P. J. and R. Medina-Mora (1995). "Completing the Loops." *Interfaces*.
- Dietz, J. L. G. (2006). *Enterprise Ontology. Theory and Methodology*, Springer-Verlag.
- Engeström, Y. (1991). *Developmental Work Research: Reconstructing Expertise Through Expansive Learning*. Human Jobs and Computer Interfaces Conference. University of Tampere.
- Fillmore, C. H. J. (1968). *The Case for Case. Universals in Linguistic Theory*. E. Bach and R. T. Harms. London, New York, Sydney, Toronto, Holt, Rinehart and Winston: 1-90.
- Fillmore, C. H. J. (1977). *The Case for Case Responded. Syntax and Semantics: 8. Grammatical relations*. P. Cole and G. M. Sadock. New York, Academic Press: 59-81.
- Frye, D. W. and T. R. Gullede (2007). "End-to-end Business Process Scenarios." *Industrial Management and Data Systems* **107**(6): 749-761.
- Goldkuhl, G. (1996). *Generic Business Frameworks and Action Modeling*. First International Workshop on Communication Modeling. Tilburg, The Netherlands.
- Goldkuhl, G. (2001). *Communicative vs Material Actions: Instrumentality, Sociality and Comprehensibility*. International Working Conference on the Language-Action Perspective on Communication Modelling (LAP'01). Montreal, Canada.

- Goldkuhl, G. and P. J. Ågerfalk (1998). Action within Information Systems: Outline of a Requirements Engineering Method. 4th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'98). Pisa, Italy.
- Gray, J. and A. Reuter (1993). Transaction Processing: Concepts and Techniques, Morgan Kaufmann Publishers, Inc.
- Hall, P., J. Owlett, et al. (1976). Relations and Entities. IFIP Working Conference on Modeling in Data Base Management Systems.
- Halpin, T. (2001). Integrating Fact-Oriented Modeling with Object-Oriented Modeling. Information Modeling in the New Millennium. M. Rossi and K. Siau, Idea Group Publishing.
- Hammer, H. and D. McLeod (1981). "Database Description with SDM. A Semantic Database Model." ACM Transactions of Database Systems **6**(3): 351-386.
- Jackson, M. (1983). System Development, Prentice Hall International.
- Jacobson, I., G. Booch, et al. (1999). The Unified Software Development Process, Addison-Wesley.
- Kappel, G. and M. Schrefl (1991). Object/Behavior Diagrams. 7th International Conference on Data Engineering. Kobe, Japan, Computer Society Press.
- Keller, G. and T. Teufel (1998). SAP R/3 Process-Oriented Implementation, Addison-Wesley Longman Limited.
- Kimball, R. (1996). The Data Warehouse Toolkit.
- Leontiev, A. N. (1978). Activity, Consciousness, and Personality. Englewood Cliffs: NJ, Prentice-Hall.
- Lübke, D., T. Luecke, et al. (2006). Using Event-Driven Process Chains for Model-Driven Development of Business Applications. XML Integration and Transformation for Business Process Management (GI-Workshop XML4BPM'06). Passau, Germany.
- Lundeberg, M., G. Goldkuhl, et al. (1978). Systemeering, Studentlitteratur.
- Mathiassen, L. (1984). Systems Development and Systems Development Method (in Danish), Aarhus University, Denmark.
- Mathiassen, L., A. Munk-Madsen, et al. (2000). Object-Oriented Analysis and Design, Marko.
- McGinnis, L. F. (2007). "Enterprise Modeling and Enterprise Transformation." Information Knowledge Systems Management **6**: 123-143.
- Mylopoulos, J. (1992). Conceptual Modeling and Telos. Conceptual Modeling, Databases, and CASE. An Integrated View of Information Systems. P. Loucopoulos and R. Zicari, John Wiley & Sons, Inc.
- Mylopoulos, J., A. Borgida, et al. (1990). "Telos: Representing Knowledge About Information Systems." ACM Transactions on Information systems.
- Nurcan, S., A. Etien, et al. (2005). "A Strategy Driven Business Process Modeling Approach." Business Process Management Journal **11**(6): 628-649.
- Orman, L. (1988). "Functional Development of Database Applications." IEEE Transactions on Software Engineering **14**(9).
- Poels, G., A. Maes, et al. (2007). "The Pragmatic Quality of Resources-Events-Agents Diagrams: An Experimental Evaluation." Information Systems Journal.
- Raghu, T. S., B. Jayaraman, et al. (2004). "Toward an Integration of Agent- and Activity-Centric Approaches in Organizational Process Modeling: Incorporating Incentive Mechanisms." Information Systems Research **15**(4): 316-335.
- Rosenquist, C. J. (1982). "Entity Life Cycle Models and Their Applicability to Information Systems Development Life Cycles." Computer Journal **25**(3): 307-315.
- Rumbaugh, J., I. Jacobson, et al. (1999). The Unified Modeling Language Reference Manual, Addison-Wesley.
- Shipman, D. (1981). "The Functional Model and the Data Language DAPLEX." ACM Transactions on Database Systems.
- Siau, K. and M. Rossi (2007). "Evaluation Techniques for Systems Analysis and Design Modelling Methods – A review and Comparative Analysis." Information Systems Journal.
- Sun, S. X., J. L. Zhao, et al. (2006). "Formulating the Data-Flow Perspective for Business Process Management." Information Systems Research **17**(4): 374-391.
- Thomas, S. J. and P. C. Fischer (1986). Nested Relational Structures. Advances in Computing Research. P. C. Kanellakis, JAI Press: 269-307.
- Turetken, O. and D. Schuff (2007). "The Impact of Context-Aware Fisheye Models on Understanding Business Processes: An Empirical Study of Data Flow Diagrams." Information and Management **44**: 40-52.
- Wegmann, A., L.-S. Lê, et al. (2007). "Enterprise Modeling Using the Foundation Concepts of the RM-ODP ISO/ITU Standard." Information Systems and e-Business Management **5**: 397-413.

- White, S. A. (2004). Introduction to BPMN, BPMN.ORG.
- Widom, J. and S. Ceri (1996). Active Database Systems: Triggers and Rules for Advanced Database Processing, Morgan Kaufmann.
- Winograd, T. and F. Flores (1986). Understanding Computers and Cognition, Addison-Wesley.
- Yourdon, E. (1989). Modern Structured Analysis. Englewood Cliffs, New Jersey, Prentice-Hall.
- Yourdon, E. (2003). Yourdon Systems Method: Model-Driven Systems Development. Englewood Cliffs, New Jersey, Yourdon Press.
- Ziemann, J., T. Matheis, et al. (2007). Modeling of Cross-Organizational Business Processes. 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'07). St. Goar, Germany.

## Working Papers from Informatics Research Group

- I-2008-03      Lars Bækgaard: Event-Based Conceptual Modeling.
- I-2008-02      Sune Dueholm Müller, Lars Mathiassen & Hans Henrik Balshøj: Organizational Change Perspectives on Software Process Improvement.
- I-2008-01      Sune Dueholm Müller, Pernille Kræmmergaard, Lars Mathiassen: Managing Cultural Variation in Software Process Improvement: A Comparison of Methods for Subculture Assessment.
- I-2006-02      Charles Møller: The Conceptual Framework for Business Process Innovation - Towards a Research Program on Global Supply Chain Intelligence.
- I-2006-01      Charles Møller: Business Process Innovation using the Process Innovation Laboratory.
- I-2005-03      Charles Møller, Pernille Kræmmergaard & Pall Rikhardsson: The Emergence of Enterprise Systems Management – A Challenge to the IS Curriculum.
- I-2005-02      Pernille Kræmmergaard & Jeremy Rose: Managing the ERP implementation journey – change in discourse from classical IT project to technology-driven organisational change initiative.
- WP 05-1        Tina Blegind Jensen: Nurses' Perception of an ERP Implementation Process – Based on a Means-End Chain Approach.



Handelshøjskolen i Århus

Aarhus  
School of Business

ISBN 9788778823137

Department of Business Studies

Aarhus School of Business  
University of Aarhus  
Fuglesangs Allé 4  
DK-8210 Aarhus V - Denmark

Tel. +45 89 48 66 88  
Fax +45 86 15 01 88

[www.asb.dk](http://www.asb.dk)